

CS3025 Compiladores

Semana 1: Contexto y Anuncios

Igor Siveroni

15 Agosto 2023

Anuncios

- La proxima semana tendremos una cuenta Discord – otra alternativa?
- Mezclaremos las horas de laboratorio con teoria y ejercicios. Quizzes seran anunciados, como minimo, con una semana de anticipacion.
- Proyecto Final: Grupos de 3, maximo.
- Proyectos intermedios. TBA.
- En la medida de lo posible, indicare lecturas y fuentes externas que esten alineadas con el contenido del curso.

Que nos falta?

Apuntes de Clase (Lecture Notes) y Guia de Laboratorio

Me gustaria usar nuestras clases para el desarrollo de los apuntes de clase y guia de laboratorio del Curso de Compiladores.

Esto necesitara de su Feedback:

- que funciona y que no.
- Algun tema que necesite incluirse o desarrollarse mas

Por esto, posiblemente veamos el mismo topico repetido pero diferente. Esto es bueno.

“Laboratorio” semana 1

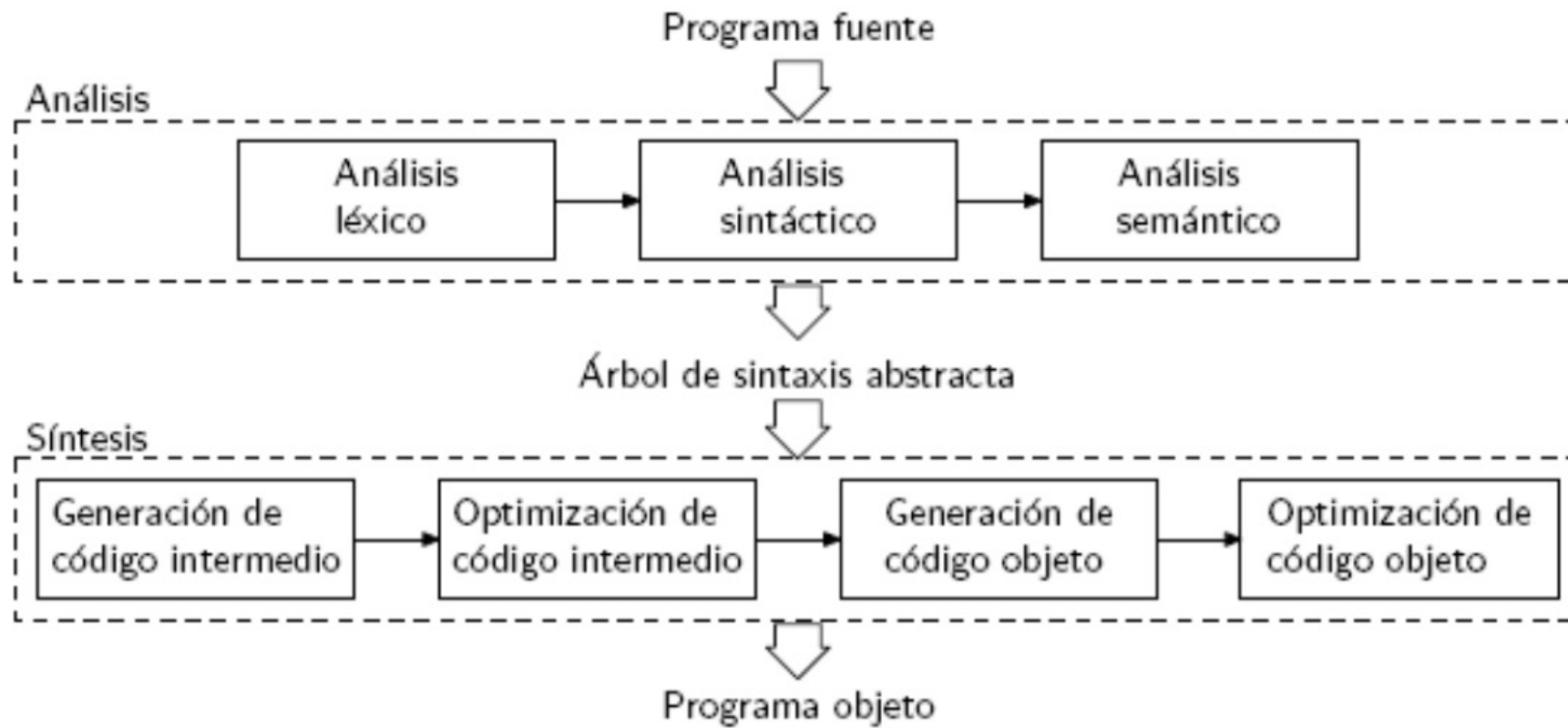
El objetivo del Laboratorio es presentar un Ejemplo de un interprete y compilador de un lenguaje fuente y objetivo sencillos – el codigo ocupa un solo archivo.

No pretendo que entiendan todo el codigo pero si despertar su curiosidad (no temor!)

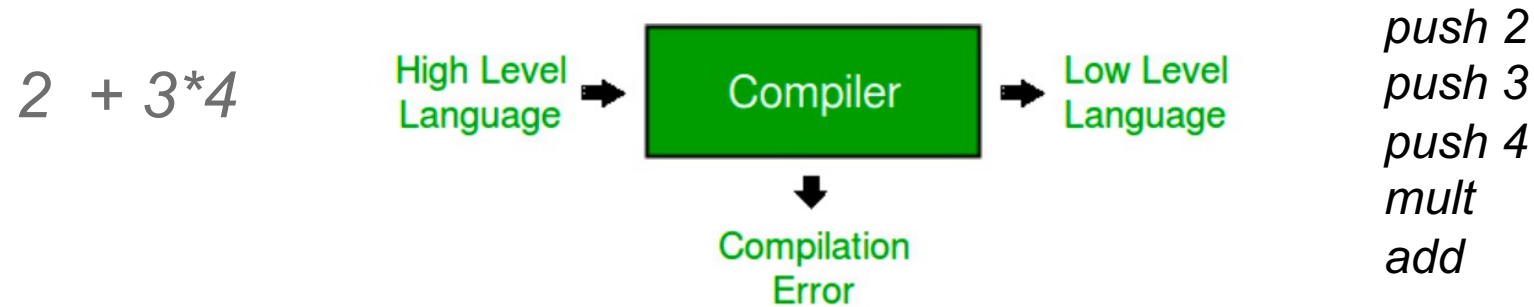
El objetivo de las primeras semanas del curso sera convertir al codigo de este labotario en algo que “tenga sentido”. Posiblemente versiones modificadas de este codigo reapareceran en el transcurso del curso.

Dicho esto ... ire con calma.

Laboratorio 1: Que partes se implementaron?



Laboratorio 1:



- Implementaciones de: Analisis Lexico (Scanner), Analisis Sintactico (Parser), interprete y compilador.
- Lenguaje Fuente: Expresiones Aritmeticas
- Lenguaje Objetivo: Lenguaje de Maquina de Pila

Es esencial entender a profundidad ambos lenguajes
Nos falta esto con el Lenguaje de Maquina de Pila

Language de Maquina de Pila

- Operaciones se realizan con elementos en la pila.
- Resultados se dejan en la pila.
- Operaciones implementadas:
 - Aritmeticas (add, subtr, mult, div): Remueven los 2 ultimos elements de la pila, se usan para la operacion aritmetica y el resulta se coloca arriba de la pila
 - push n: Poner n arriba de la pila
 - pop: remover el ultimo element de la pila
 - dup: Repetir / copiar el ultimo elemento de la fila

Esto nos basta para evaluar nuestras expresiones aritmeticas.
Con la excepcion de $\text{pow}(a,b)$

Stack Machine: Ejemplo 1 “2 + 3*4”

Code:	Stack:
push 2	2
push 3	2 3
push 4	2 3 4
mult	2 12
add	14

Laboratorio 1: continuacion

- Compilar los dos nuevos programas:
 - `g++ -o compiler compiler.cpp`
 - `g++ -o vm svm.cpp`
- Run them:
 - `./compiler "2 * 3*4" ejemplo1.sm`
 - `./ vm ejemplo1.sm`

Obtienen el mismo resultado

Pregunta / Lecturas

Pregunta: Que necesitamos para implementar algo como $\text{pow}(2,n)$?

Lectura (ver semana 2):

Capitulo 2, Modern Compiler Implementation in Java, A. Appel.

(lo subire en un par de horas)